UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.
08/789,025	01/27/97	KERZMAN J	33012/184/10

LM01/0628

CHARLES A JOHNSON
UNISYS CORPORATION
MS 4772
2276 HIGHCREST ROAD
ROSEVILLE MN 55113

EXAMINER	
JONES, H	
ART UNIT	**PAPER NUMBER**
2763	6

DATE MAILED: 06/28/00

Please find below and/or attached an Office communication concerning this application or proceeding.

Commissioner of Patents and Trademarks

Office Action Summary

Application No.	Applicant(s)
98/789,025	Kerzman et al.
Examiner	**Group Art Unit**
Hugh Jones	2763

☒ Responsive to communication(s) filed on *May 26, 1000*

☐ This action is **FINAL.**

☐ Since this application is in condition for allowance except for formal matters, **prosecution as to the merits is closed** in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11; 453 O.G. 213.

A shortened statutory period for response to this action is set to expire *3* month(s), or thirty days, whichever is longer, from the mailing date of this communication. Failure to respond within the period for response will cause the application to become abandoned. (35 U.S.C. § 133). Extensions of time may be obtained under the provisions of 37 CFR 1.136(a).

Disposition of Claim

☒ Claim(s) *1-41* is/are pending in the applicat

Of the above, claim(s) _____ is/are withdrawn from consideration

☐ Claim(s) _____ is/are allowed.

☒ Claim(s) *1-41* is/are rejected.

☐ Claim(s) _____ is/are objected to.

☐ Claims _____ are subject to restriction or election requirement.

Application Papers

☐ See the attached Notice of Draftsperson's Patent Drawing Review, PTO-948.

☐ The drawing(s) filed on _____ is/are objected to by the Examiner.

☐ The proposed drawing correction, filed on _____ is ☐ approved ☐ disapproved.

☒ The specification is objected to by the Examiner.

☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. § 119

☐ Acknowledgement is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d).

☐ All ☐ Some* ☒ None of the CERTIFIED copies of the priority documents have been

☐ received.

☐ received in Application No. (Series Code/Serial Number) _____ .

☐ received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

*Certified copies not received: _____

☐ Acknowledgement is made of a claim for domestic priority under 35 U.S.C. § 119(e).

Attachment(s)

☒ Notice of References Cited, PTO-892

☐ Information Disclosure Statement(s), PTO-1449, Paper No(s). _____

☐ Interview Summary, PTO-413

☐ Notice of Draftsperson's Patent Drawing Review, PTO-948

☐ Notice of Informal Patent Application, PTO-152

— SEE OFFICE ACTION ON THE FOLLOWING PAGES —

DETAILED ACTION

Introduction

1. Examiner has carefully reviewed the history of prosecution of this and all co-pending applications. The prior art rejections are maintained and expanded. Other issues including *112 problems* and *improper incorporation by reference* have been detected by the Examiner - consequently, this action is non-final.

Specification

2. Applicant has requested that Examiner review incorporated co-pending applications including all art contained therein. As Applicant can appreciate, this places an extreme burden on the Examiner *(as well as all other Examiners)* because Examiners must review each application. In the interest of compact prosecution, the Examiner would be appreciative if the Applicant would kindly supply copies of the indicated applications. **Examiner also requests a *claim matrix* of all applications which Representative considers related so that Examiner can review all applications for possible double patenting.**

3. **The attempt to incorporate subject matter into this application by reference to U. S. applications listed on pages 1-2 of the specification is improper because these applications also incorporate essential matter by reference.** The following is recited from section 608.01(p) of the MPEP:

"INCORPORATION BY REFERENCE

The Commissioner has considerable discretion in determining what may or may not be incorporated by reference in a patent application. General Electric Co. v. Brenner, 407 F.2d 1258, 159 USPQ 335 (D.C. Cir. 1968). The incorporation by reference practice with respect to applications which issue as U.S. patents provides the public with a patent disclosure which minimizes the public's burden to search for and obtain copies of documents incorporated by reference which may not be readily available. Through the Office's incorporation by reference policy, the Office ensures that reasonably complete disclosures are published as U.S. patents. The following is the manner in which the Commissioner has elected to exercise that discretion. Section A provides the guidance for incorporation by reference in applications which are to issue as U.S. patents. Section B provides guidance for incorporation by reference in benefit applications; i.e., those domestic (35 U.S.C. 120) or foreign (35 U.S.C. 119(a)) applications relied on to establish an earlier effective filing date.
A. Review of Applications Which Are To Issue as Patents.

An application as filed must be complete in itself in order to comply with 35 U.S.C. 112. Material nevertheless may be incorporated by reference, Ex parte Schwarze, 151 USPQ 426 (Bd. App. 1966). An application for a patent when filed may incorporate "essential material" by reference to (1) a U.S. patent or (2) a pending U.S. application, subject to the conditions set forth below.

"Essential material" is defined as that which is necessary to (1) describe the claimed invention, (2) provide an enabling disclosure of the claimed invention, or (3) describe the best mode (35 U.S.C. 112). ***In any application which is to issue as a U.S. patent, essential material may not be incorporated by reference to*** *(1) patents or applications published by foreign countries or a regional patent office, (2) non-patent publications,* ***(3) a U.S. patent or application which itself incorporates "essential material" by reference****, or (4) a foreign application.*

Nonessential subject matter may be incorporated by reference to (1) patents or applications published by the United States or foreign countries or regional patent offices, (2) prior filed, commonly owned U.S. applications, or (3) non-patent publications however, hyperlinks and/or other forms of browser executable code cannot be incorporated by reference. See MPEP § 608.01. Nonessential subject matter is subject matter referred to for purposes of indicating the background of the invention or illustrating the state of the art.

1. Complete Disclosure Filed

If an application is filed with a complete disclosure, essential material may be canceled by amendment and may be substituted by reference to a U.S. patent or an earlier filed pending U.S. application. The amendment must be accompanied by an affidavit or declaration signed by the applicant, or a practitioner representing the applicant, stating that the material canceled from the application is the same material that has been incorporated by reference.

If an application as filed incorporates essential material by reference to a U.S. patent or a pending and commonly owned U.S. application, applicant may be required prior to examination to furnish the Office with a copy of the referenced material together with an affidavit or declaration executed by the applicant, or a practitioner representing the applicant, stating that the copy consists of the same material incorporated by reference in the referencing application. However, if a copy of a printed U.S. patent is furnished, no affidavit or declaration is required.

Prior to allowance of an application that incorporates essential material by reference to a pending U.S. application, the examiner shall determine if the referenced application has issued as a patent. If the referenced application has issued as a patent, the examiner shall enter the U.S. Patent No. of the referenced application in the specification of the referencing application (see MPEP § 1302.04). *If the referenced application has **not issued as a patent, applicant will be required to amend the disclosure of the referencing application to include the material incorporated by reference.** The amendment must be accompanied by an affidavit or declaration executed by the applicant, or a practitioner representing the applicant, stating the amendatory material consists of the same material incorporated by reference in the referencing application.*

4. The disclosure is objected to because of the following informalities: the reference to Application numbers and Attorney docket numbers throughout the specification is objected to. Appropriate correction is required.

Claim Rejections - 35 USC § 112

5. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. Claims 1-41 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. *Examples are provided for Representative. Representative is responsible for locating and correcting all other and similar instances. Examples include*:

- (claim 1): *generating*: how? By who?

- (claim 1): *selecting*: how? By who?

- (claim 1): *establishing* how?

- (claim 1): *hot keys*: presumably *function keys* - what kind?

- (claim 2): *options determine the display*: what options? How do they determine the display?

- (claim 12) what kind of *physical violations*?

- (claim 13) what is a *parking lot violation*?

- (claim 14): What is an *off-grid error*?

- (claim 15): what is an *out-of-context cell*?

- (claim 16): *identify* how?

Claim Rejections - 35 USC § 102

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless --
>
> (e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

8. **Claims 1-41 are rejected under 35 U.S.C. 102(e) as being clearly anticipated by**

Nishiyama (of record) or Beausang et al. or Yano et al. or Dangelo et al. (U. S. Patent 5,555,201).

9. Nishiyana discloses (abstract):

"A computer implemented process and system for providing a scan insertion process having a reduced set of constraint driven compiler optimizations that provide an efficient and effective optimization for design for test implementations. The present invention includes a three tiered effort performance optimization process within a scan insertion process; a first tier operates to perform a set of optimizations (size design) only on elements of the design added for design for test (DFT). The second tier offers the first tier and performs the size design optimizations across all of the design while the third tier offers the second tier with sequential optimizations, circuit size downs, and another size design. *Each higher user-selectable tier offers more complex optimizations and consumes additional processing time. An option to perform design constraints optimization (max fanout, max signal transition, and max capacitance) is also available. By utilizing a reduced set of performance optimizations, the present invention offers a post scan insertion compile technique that is fast enough to be practically used on chip level netlists. Hierarchical compilations for DFT are therefore allowed.* Since the modified scan insertion procedure can operate in conjunction with a TR compiler of the present invention, the modified scan insertion procedure breaks loopback connections and generates proper scan chains. The scan insertion process of the present invention is compatible with netlists that contain a mixture of scan cells and non-scan cells."

Ccol. 6, line 11 to col. 7, line 7 discloses:

"FIG. 1 shows a schematic generator for automatically generating a logic circuit as an embodiment according to the present invention. This system includes a net information file 1 storing therein information about elements (including circuit parts) and connective or connecting information thereof, a contour information file 2 for storing therein contour or shape information of drawing symbols representing respective elements, a logic diagram information file 3 for storing therein drawing information about placements or placements and wirings or routings of a logic circuit diagram (also called a logic diagram), a cell library 4 storing therein contour information and functional information of each element, logic diagram drawing section 5 for receiving as an input thereto the logic diagram information to draw a logic circuit diagram by use of a drawing terminal 51 such as a display or a plotter, and a placement/routing processing section 11-16.

For example, the net information file 1 keeps a net list (connective information) or the like of a logic circuit obtained as an output from an automatic logic synthesizing system or an automatic circuit

transforming system. Information stored in the net information file 1 may also be employed for a logical vertification, a timing check, an automatic layout system, etc.

In the element level assignment 11, based on the number of logical stages from an input port (input terminal) of each of the elements and the circuit parts contained in the net list kept in the net information file 1, a level is determined for each element and for each circuit part.

In the reference level position determination 12, the system determines positional relationships in a direction orthogonal to a level direction between the elements or circuit parts at the reference level (namely, sequential relationships for placing or arranging elements or circuit parts in a direction vertical to the level direction). In this embodiment, for simplification, the maximum level (i.e. a level at which an output port exists) is set as the reference level; however, even if a level at which an input port exists (namely, level 0) is set as the reference or elements and circuit parts at a plurality of levels are assigned with the reference level, there does not occur any problem.

In the relative placement at each level 13, based on sequential relationships of positions of the elements (i.e. components or circuit parts) at the reference level, positional relationships of positions are determined for elements at the next level (in this embodiment, at the next level in the descending order), thereby determining sequential relationships of positions for all elements of the logic circuit.

In the placement processing 11 to 13, few considerations have been given to the contour of each placement element (namely, to the contour of each drawing symbol of the elements and circuit parts). In consequence, assuming that the level of each element determined through the placement steps 11 to 13 is a first (global) coordinate and that a sequence obtained from sequential relationships of positions also determined through the placement steps 11 to 13 is a second (global) coordinate, graphic placement (not related to the contour) is determined for each element.

Incidentally, in this example, the sequential relationships of positions at each level are determined in a descending order. However, when the reference level is set to the level 0, the relationships are to be determined in the ascending level order; whereas in other cases, the relationships are to be determined in both directions including the ascending and descending level rders."

See: fig. 1, 6; col. 1-4.

10. Beausang et al. disclose (col. 8, lines 1-25):

"Specific aspects of the present invention are operable within a programmed computer aided design (CAD) system. A CAD system operable to implement the elements of the present invention is shown in FIG. 2. In general, the CAD system of the present invention includes a computer system 112 which includes a bus 100 for communicating information including address, data, and control signals, a central processor 101 coupled with the bus 100 for processing information and instructions,

a random access memory 102 coupled with the bus 100 for storing information and instructions for the central processor 101, a read only memory 103 coupled with the bus 100 for storing static information and instructions for the processor 101, a data storage device 104 such as a magnetic or optical disk and disk drive coupled with the bus 100 for storing information and instructions, a display device 105 coupled to the bus 100 for displaying information to the computer user, *an alphanumeric input device 106 including alphanumeric and function keys coupled to the bus 100 for communicating information and command selections to the central processor 101, a cursor control device 107 coupled to the bus for communicating user input information and command selections to the central processor 101,* and a signal generating device 108 coupled to the bus 100 for communicating signals that are input and output from the system 112."

Col. 13, line 27 to col. 14, line 52 disclose:

"FIG. 8 illustrates an overall flow diagram of a synthesis process 600 in accordance with embodiments of the present invention and its logic blocks are implemented within the computer controlled CAD system described above. Flow 600 includes the TR compiler (test ready compile) 625 of the present invention and the modified scan insertion procedure 645 (constraint driven scan insertion) of the present invention. Flow 600 also includes a generic HDL compiler 615 which will be described below but is different from the TR compiler 625 of the present invention. Generic compiler 615 is analogous to generic compiler 203.

The flow 600 receives an HDL description 605 of an integrated circuit layout along with a set of design constraints 610 (including design rule limitations, and performance limitations, such as area, timing, power, etc.) that are pertinent to the circuit described by the HDL description 605. Design rules as used herein refer to maximum fanout, maximum signal transition time, and maximum node capacitance. The HDL description 605 can be of a number of different formats, such as VHDL or Verilog and can also represent an entire IC design, but typically represents a module of the overall IC design. The HDL description 605 can be stored in a computer memory unit (e.g., unit 102 or 104) and is fed into an optional generic compiler logic block 615 that is well known in the art. This generic compiler 615 transforms the HDL description 605 into a technology independent netlist 620 that is more readily recognized by the TR compiler 625. Block 615 performs a process on the input netlist 620 to generate a technology independent or generic netlist of the IC layout by interfacing with a synthetic library or "designware" library. Technology independent netlist 620 is composed of logical primitives and operators of the IC layout but the components described therein contain no structure. The generic compile process 615 and resultant output 620 are well known in the art.

The netlist 620 of FIG. 8 generated by block 615 is input to the TR compiler logic block 625 of the present invention. The TR compiler 625 is described in more detail in FIG. 9 which illustrates the particular stages within the TR compiler 625 where the sequential circuits are replaced and where the loopback connections are inserted. The TR compiler 625 in FIG. 8 performs a process on the

generic netlist that interfaces with a technology specific library, "cell" library, so that a mapped netlist can be generated that includes specific structural information about the components used in the compiled design. In addition, the TR compiler 625 of the present invention: (1) replaces the HDL memory cells specified in or inferred from the netlist 620 (e.g., non-scan cells) with scannable memory cells; and (2) inserts loopback connections in each scannable memory cell added. FIG. 5A, FIG. 5B, FIG. 5C and FIG. 5D illustrate the memory cell replacements performed by the present invention while FIG. 6A and FIG. 6B illustrate the addition of the loopback connections performed by the present invention TR compiler 625.

Referring to FIG. 8, by performing the above processes, the TR compiler 625 of the present invention is able to better optimize for the eventual construction and completion of the DFT circuitry. In this way, it is more likely that the resultant test circuit design will meet constraints 610. The output of the TR compiler 625 of the present invention is a non-scannable technology dependent netlist 630 that comprises scannable memory cells with loopback connections 440. Although called "nonscannable" because of the loopback connections, netlist 630 is nevertheless a fully scanned netlist in that the TR compiler 625 replaced each HDL specified or inferred sequential cell (e.g., non-scan cell) by an equivalent scannable cell. This netlist 630 can be stored in a memory unit of the computer system such as RAM 102 or the storage device 104 (FIG. 2). The overall non-scannable netlist 630 is optimized to constraints 610 and is a gate level mapped netlist and therefore is technology specific.

The non-scannable netlist 630 of FIG. 8 is then input to a DFT design rule checker logic block 635 ("DRC"). Any of a number of well known DRC processes can operate within the present invention including a DRC as described by E. B. Pitty, D. Martin, and H. T. Ma in a paper entitled "A Simulation-Based Protocol-Driven Scan Test Design Rule Checker," published in IEEE International Test Conference, page 999, paper 40.2 (1994). The DRC 635 checks the scannable memory cells in the netlist 630 to determine which cells should be violated according to the discussion herein with respect to FIG. 4. Processing flows to DRC block 635' where those scannable memory cells that are determined not to be part of a scan chain by DRC block 635 are marked as violated by the DRC at logic block 635'. Cells marked as violated will be unscanned by the modified scan insertion process 645. The act of unscanning replaces the violated memory cells with an equivalent non-scan memory cell. When unscanned, the loopback connection 440 associated with a violated cell is also destroyed. When violated, the memory cell is referred to as unscannable."

See, also: fig. 1-4, 8-12, 20; col. 1-5.

11. Yano et al. disclose (abstract):

"The semiconductor integrated circuit enjoys a high performance and can be produced at a low production cost and within a short time. A cell has an internal circuit connection such that an output terminal is connected to a plurality of input terminals through source-drain paths of active

devices connected in the tree form, and gate electrodes of the active devices are connected to other input terminals. Two such cells having the same internal circuit connection, the same disposition of the internal circuit devices and the same disposition of the input/output terminals are disposed on the same chip, and mutually different logics can be accomplished by changing the form of application of input signals from outside the cells to the input terminals. A chip area of an integrated circuit designed by CAD using a cell library can be reduced and a high speed circuit operation can be attained. The present invention provides remarkable effect for improving performance of an ASIC, a microprocessor, etc., and for reducing the cost of production."

Col. 12, line 23 to col. 14, line 4 disclose:

"As described above, the cell registered to the cell library of CAD is a logic circuit for which layout has already been made, and is prepared before the logic design of the entire integrated circuit. Since layout of a cell is time-consuming and troublesome, it is quite natural to constitute a cell library by selecting the cells having logic functions which have high frequency of use. Conventionally, the logic functions having high frequency of use are a 1-input inverter, 2- or 3-input AND, OR, XOR (or their negation), and it is a highlight scene for a designer of the logic how to efficiently constitute complicated logics of integrated circuit by combining them.

In contrast, when the logic output (22) of the PC3 cell according to this embodiment shown in FIG. 1 is expressed as a function of the signal of the input terminals (5 to 21) by the Boolean formula, it becomes much more complicated as shown below (see "logic function" in FIG. 1):

(22) (((19)(18)+(20)(17))(16)+(21)(15)))N

Accordingly, it would be considerably difficult for those skilled in the art to dare to use those circuits have such complicated logic functions and frequency of use of which is believed low, as the basic cell of the cell library. In other words, because it takes a great deal of time and labor to constitute the cell library, registration of the circuits hardly having high frequency of use cannot be made in the conventional logic design unless extremely high motivation exists.

The Pasternak et al reference cites AND, OR, XOR as the logic functions of the standard cell in line with this conventional, traditional concept. The Yano et al reference, too, is based on this traditional concept. Yano is one of the co-inventors of the present invention but in 1990 when this reference was published, he was aware of the fact that the AND circuit could be changed to the OR circuit by merely changing partially the signal application connection of an internal circuit of a bifurcate pass transistor circuit and therefore described this concept in the reference described above. However, because the change of connection was necessary, though partially, he was of the opinion that an AND cell and an OR cell were separately necessary, and he made no doubt about the conventional premise that logic design was made on the basis of the separate cells such as AND, OR

and XOR. In this way, it was extremely customary for designers of logic circuits to make logic design using the separate cells such as AND, OR and XOR and this premise was like the premise that arithmetic calculation was made by using numerals. In other words, it has been extremely difficult for those skilled in the art to again look at this traditional concept.

In contrast, the inventors of the present invention have found out that a large number of different logic functions can be accomplished by using only one kind of the cell PC3 shown in FIG. 1 and by changing the forms of application of input signals from outside the cell. As a result, the present inventors have got rid of the conventional fixed idea that the cell function must be comprehensible on the basis of AND and OR, and have reached the concept that this bifurcate connection circuit itself should be registered as the cell and an ideal logic design must be reconstituted on the basis of such a cell,

On the other hand, a plurality of cells PC3 having different logic functions and shown in FIG. 1 have the same internal circuit connection and the same layout pattern and only the forms of application of the input signals from outside the cells are different. Accordingly, the feature that the functions of the cell PC3 shown in FIG. 1 are complicated and are not easily comprehensive would have been a critical problem several years ago. Even though the cell PC3 was prepared for the cell library, the designer of logic would not have attempted to use such an incomprehensive cell.

Recently, however, logic automatic synthesis tools (those tools which automatically output a connection net list of cells accomplishing an intended logic function when such a logic function is inputted) have been drastically put into practical application, and optimum circuit design of logic circuits (that is, decision of the connection relation of the cells) has been made by a computer but not by the designer. Under such circumstances, the present inventors have come to realize that whether or not the cell function is comprehensive to the designer has not been latently important any longer. The present inventors have completed on the basis of this concept the present invention which overthrows the basis of the logic design of the integrated circuits using AND, OR, XOR, INVERT as the basic cells that have long been employed. As a matter of fact, the present inventors have succeeded in the development of softwares which accomplish arbitrary logic functions by combining the cells shown in FIG. 1. Also, the present inventors have confirmed that when such softwares are used, the area, speed and power consumption of the integrated circuit can be drastically improved."

See, also fig. 1, 3, 7-8, 23-24; col. 2-7.

12. Dangelo et al. disclose (Col. 6, line 38 to col. 12, line 67 disclose):

"The present invention provides a method and system for hierarchical display of control and dataflow graphs allowing a user to view hierarchically filtered control and dataflow information related to a design. The invention employs information inherent in the design description and information derived from design synthesis to identify "modules" of the design and design hierarchy.

The user can specify a level of detail to be displayed for any design element or group of design elements. Any CDFG (control and dataflow graph) object can be "annotated" with a visual attribute or with text to indicate information about the design elements represented by the object. For example, block size, interior color, border color, line thickness, line style, etc., can be used to convey quantitative or qualitative information about a CDFG object.

Examples of information which can be used to "annotated" objects include power dissipation, propagation delay, the number of HDL statement represented, circuit area, number of logic gates, etc. The user is able to expand and/or compress CDFG blocks either "in-place" on a higher level CDFG display or to be displayed in isolation. Simulation-related data can also be used to annotate the CDFG.

By viewing CDPG's (particularly annotated CDFG's) for a variety of trial designs, a problem-solving user can gain quick insight into the effects and effectiveness of various design choices.

It is therefore an object of the present invention to provide an improved ECAD system whereby the characteristics of schematic editor, schematic compiler, and simulator are all presented to the user in a fashion such that they appear as a single, integrated function.

It is a further object of the present invention to allow portions of a circuit which is being designed on such an improved ECAD system to be simulated in isolation without requiring that those circuit portions be copied to another schematic, regardless of whether or not the overall schematic diagram has been completed, and regardless of whether or not the circuit portion has other connections.

It is a further object of the present invention to allow the user to view full or partial simulation results on the display screen representation of the schematic as it is being edited on the improved ECAD system.

It is a further object of the invention to enable the user to view state, performance, loading, drive strength or other relevant data (hereinafter, "state" data) in display areas immediately adjacent to the schematic object to which it pertains.

It is a further object of the invention to enable the user to perform simulator setup on the schematic diagram by using point and select techniques to identify items to be simulated, input values, override values, and points to be monitored.

It is a further object of the invention to enable the user to create state tables for circuits, portions of circuits, or components.

It is a further object of the invention to enable the user to store the interactive state data for viewing at another time.

It is a further object of the invention to enable the user to create macros to move through the simulation in defined steps (of "n" time units of the lowest system granularity), or to cycle a clock.

It is a further object of the invention to enable the user to pop up data sheets or any library element being used, and further to allow the user to define his/her own data sheets and to allow these to be popped up in the schematic editor environment.

It is a further object of the present invention to provide a methodology for deriving a valid structural description of a circuit or system from a behavioral description thereof, thereby allowing a designer to work at higher levels of abstraction and with larger, more complex circuits and systems.

It is a further object of the present invention to provide a technique for automatically translating behavioral descriptions of a circuit or system into physical implementations thereof.

It is further object of the invention to raise the level of design validation from a structural (net list) level to a behavioral level.

It is a further object of the invention to provide a more standardized design environment, thereby alleviating the need for cross-training between different design platforms and allowing resources to be directed more towards synthesis and testability.

It is a further object of the invention to provide a technique interactive design, synthesis, simulation and graphical display of electronic systems.

It is a further object of the present invention to provide a technique for indicating the source of design rule violations in a logic synthesis or design process to a user.

It is a further object of the present invention to provide a technique for automatically providing suggestions to a user about possible alterations or corrections to a design or design description of an electronic system which violates design rules.

It is a further object of the invention to accomplish the above objects in a manner compatible with the design of board-level systems, multi-chip modules, integrated circuit chips, and ASICs using core modules.

It is a further object of the invention to accomplish the above objects independently of scale, complexity, or form factor of the electronic system.

It is an additional object of the present invention to provide a technique for gathering and condensing relevant information about an electronic design into a form readily accessible to the designer, for presenting this information to the designer in a readily digestible (e.g., visual) format, and to allow the designer to interact with this information in a facile, ergonomic manner.

It is an additional object of the present invention to provide a technique for propagating information from level-to-level in a hierarchical design system.

It is an additional object of the present invention to provide a technique for providing the designer with suggestions regarding the steps necessary to create information which is not available at a particular instance of the design process.

According to the invention, there is provided an electronic CAD system operated with a suite of software tools for enabling a designer to create and validate a structural description and physical implementation of a circuit or system (hereinafter, "device") from a behavior-oriented description using a high-level computer language. The methodology includes the following steps:

First, the designer specifies the desired behavior of the device in a high-level language, such as VHDL. The description includes high-level timing goals.

Next, in a "behavioral simulation" step, starting with the VHDL behavioral description of a design, the designer iterates through simulation and design changes until the desired behavior is obtained.

Next, in a "partitioning" step, the design is partitioned into a number of architectural blocks. This step is effectively one of exploring the "design space" of architectural choices which can implement the design behavior. Links to the physical design system enable high level timing closure by constraining the feasible architectural choices to those which meet the high-level timing and area (size) goals. This is a key step because it represents the bridge between the conceptual level and the physical level. A second function of this step is to direct the various architectural blocks to the appropriate synthesis programs.

Next, in a "logic synthesis" step, a number of separate programs are used to efficiently synthesize the different architectural blocks identified in the partitioning step. Those blocks having highly regular structures or well understood functions are directed to specific synthesis tools (e.g. memory or function compilers). Those blocks with random or unstructured logic are directed to more general logic synthesis programs. The output of this step is a net list of the design.

Next, in a "physical simulation" step, the gate-level design description is simulated, comparing the results with those from the initial behavioral simulation. This provides a check that the circuit implementation behaves as intended, and that the timing goals are achieved.

Optionally, the design is back-annotated to ensure that other physical design limitations, such as capacitive loads and parasitics, are not exceeded.

Finally the design is input to existing software systems which control the physical implementation of the design, such as in an ASIC (Application Specific Integrated Circuit) device.

An important feature of the present invention is that, as with all top-down design approaches, the foregoing is a process of architectural refinement in which design realization moves down through levels of abstraction. The characteristics of VHDL and the disclosed methodology enable this process to occur without losing the intent and meaning present at higher levels. This is the key to automating the process.

Another important feature is that the partitioning step, or partitioner, in effect, uses high-level timing information extracted from the chip floorplan to constrain the design into the feasible architectural choices which meet the high-level timing goals. These constraints are key to allowing the process to converge to specific physical embodiments.

Another important feature is that the methodology enables timing closure without going to actual layout, solving one of the most difficult problems in design automation today, namely the inability to get timing closure at even the gate level effectively which in the past has forced designers to create two designs: a logic design and a timing design. Using the methodology disclosed herein, timing closure can be obtained by using a form of back annotation which will extract timing data from floorplanning-level layouts and then incorporate this data into the I/O (Input/Output) ports of the VHDL behavioral description.

According to an aspect of the invention, the behavioral (VHDL) description of the device is interpreted by attaching one or more semantic rules to each of the syntactic rules underlying the behavioral description. This is accomplished (such as via Prolog) using a "syntax attributed tree".

According to the invention, the electronic CAD system comprises a computer processor, mass storage, a display screen, means for user input, and means for circuit simulation. The electronic hardware of the means for simulation may comprise the ECAD system's computer, one or more general purpose computers interfaced to the ECAD system's computer, one or more hardware simulators interfaced to the ECAD system's computer, or any combination of these. ***The user interacts with the ECAD system through the use of an object-oriented user interface, whereby the user may create, select, move, modify and delete objects on the display screen, where objects may represent circuit components, wires, commands, text values, or any other visual representation of data. The graphical and software techniques of interacting with a user on such an object-oriented user interface are well known to those skilled in the art and need not be elaborated upon in this discussion.***

A component database resides on the ECAD system's mass storage. This database comprises a number of data objects: graphical symbols, connection information, timing parameters, and simulation models corresponding to various electronic components. These data objects contain all of the information necessary to display, interconnect, and edit schematic symbols on a graphical display screen. The simulation model data objects contain the behavioral data corresponding to the components represented by the graphical objects such that the simulator may produce results closely approximating those that would be observed if real components were used and measured on standard laboratory instrumentation.

Five major software program functions run on the ECAD system: a schematic editor, a logic compiler, a logic simulator, a logic verifier, and a layout program. The schematic editor program allows the user of the system to enter and/or modify a schematic diagram using the display screen, generating a net-list (summary of connections between components) in the process. The logic compiler takes the net list as an input, and using the component database puts all of the information necessary for layout, verification and simulation into a schematic object file or files whose format(s) is(are) optimized specifically for those functions. The logic verifier checks the schematic for design errors, such as multiple outputs connected together, overloaded signal paths, etc., and generates error indications if any such design problems exist. The logic simulator takes the schematic object file(s) and simulation models, and generates a set of simulation results, acting on instructions initial conditions and input signal values provided to it either in the form of a file or user input. The layout program generates data from which a semiconductor chip (or a circuit board) may be laid out and produced.

These programs are typical of similar, discrete programs existing in the current art, but are slightly modified (improved in their functionality) in the source of their control information. The editor's user interface is extended such that the simulator functions may be requested by the user. It is further modified such that whenever a change is made to the schematic, the editor updates its output files (net list, etc.) and signals the logic compiler to re-compile the schematic using the new data. The logic

compiler is modified to accept such commands directly from the editor, rather than from user input. The simulator is modified such that it can accept directly from the editor: requests for simulation runs, initial data, signal data, and other information usually entered by the user via the keyboard and/or pointing device. It is further modified to signal the editor that it has completed a simulation operation, and to provide its results in the form of a data structure, either in memory or in a disk file, rather than to the display screen. The logic verifier is also modified such that it interacts with the editor directly, rather than with the display screen and keyboard/pointing device.

Further according to the invention, the editor causes the logic compiler to re-compile the schematic each time a graphical object (schematic symbol) is added, modified, or deleted, and each time a connection is made, changed or removed. In this way, the editor ensures that the net-list and simulation structures are always current and representative of the schematic diagram as displayed on the ECAD system's graphical display screen.

At any time, the user may instruct the editor to create areas on the display screen adjacent to selected schematic symbol connection points (pins) or on connection nets (wires). By conventions already in place in all editors, compilers, and simulators, these connection points and/or connection nets are uniquely identifiable. The user may specify that these data areas are to contain textual state data, or graphical state data. Next the user may identify certain signal values to be injected into the circuit representation. Ordinarily these would be input signals, but for simulation of part of the schematic, it is possible to override the outputs of selected schematic object to force special conditions to exist on a net, or to force signals into a particular input connection point on a schematic object, effectively overriding its connection. It is also possible for the user to indicate that only certain components are to be compiled and simulated, thus improving the compile and simulation times. This is accomplished by one of two means: either subset net-list and object files are created, reducing the amount of data to be handled by the logic compiler and logic simulator, or software flags are provided in the data structures indicating which data objects are to be considered active, allowing the logic compiler to selectively compile the schematic and allowing the logic simulator to selectively simulate the schematic.

All of the user input occurs by pointing with the pointing device and selecting connection nodes, nets or devices and issuing commands which affect the selected object's numerical parameters. Each data object (schematic symbol, connection net (wire), and connection point (pin)) has special parameters which allow it to be made eligible or ineligible for compile and/or simulation, and to have some or all of its other parameters overridden for the purposes of simulation.

When the user wishes to perform a simulation he issues a command which is relayed by the editor to the simulator. The simulator performs a simulation run according to the user's specification and places the simulation results into a data structure. It signals the editor that the simulation is complete and then fills in the results on the screen, according to the user's display specification. The user may specify a complete simulation run from a set of initial conditions or a simulation stepped run which continues from the last simulation's ending point. In the event of a complete simulation run, a new simulation results data structure is created and filled in. In the event of a stepped run, the

simulator appends new simulation data to the end of the previously created simulation results data structure.

Simulators, by their nature, must maintain the last state (history) of every node for every enabled component in the schematic. However, this history is kept only for those signals requested. This is done to minimize the amount of data storage required. It is possible to request that the history be maintained for all nodes at the expense of some amount of additional memory (or disk space) required.

When the editor receives notification from the simulator that the simulation run is finished, it displays the simulation data on the screen according the specifications for the display areas that the user has requested. If it is a textual display area, then the last state of the node is written into the allocated display area. If it is a graphical (timing diagram) display area, then the history data is presented in the allocated display area in the form of a timing diagram. In either case, the user can step through the state data back from the end point to any previous point in the simulation from the beginning of the session.

The editor may also create, at the user's request, an area on the screen for the presentation of a state table. The user identifies the signals to be monitored and identifies the simulation conditions. The editor then draws a table on the screen and headings corresponding to the monitored signals' names, and requests a series of stepped simulations. After each step, the editor records the last state data into columns under the signal name headings, thus creating a state table of the type seen in component specifications.

The techniques described hereinabove for electronic system synthesis, graphical design of an electronic system, simulation and display are independent of the type of electronic system being designed and may be applied with equal facility to multi-board systems, board-level designs, ASICs, custom integrated circuits, portions of systems, or multi-chip modules.

According to an aspect of the invention, the techniques of electronic circuit synthesis and simultaneous graphical editing and display can be used in combination to provide a user with means for viewing a behavioral synthesis of an electronic system in progress, and to simulate the all or a portion of the electronic system and to view signals within the electronic system.

According to another aspect of the invention, design rule violations (e.g., timing violations detected during synthesis) flagged by the synthesis process can be presented to the user by display the portion of the electronic system involved in the violation in schematic diagram form, and presenting simulation results which illustrate the violation on the schematic diagram.

According to another aspect of the invention, an expert system can be used to analyze the electronic system and the design rule violation, and to suggest to the user possible alterations or corrections to the design of the electronic system which would eliminate or correct the design rule violation.

Parent U.S. patent application Ser. No. 07/917,801, filed Jul. 20, 1992 (U.S. Pat. No. 5,220,512; Jun. 15, 1993) generally describes a system wherein information relating to signal values is displayed to the user, simultaneously and interactively, immediately adjacent a selected line on a

schematic diagram. The present invention is similarly interactive and simultaneous, and provides the user with the ability to display and edit information other than signal values in the context of a broader range of graphical objects that may be derived from various hierarchical levels of abstraction of an electronic design on an ECAD system.

According to an additional aspect of the invention, the editor may also create, at the user's request, an area on the display for the presentation of a state table. The user identifies the signals to be monitored, and identifies the simulation conditions. The editor then draws a table on the screen, wherein headings correspond to the monitored signals' names, and requests a series of stepped simulations. After each step, the editor records the last state data into columns under the signal name headings, thus creating, for example, a modified state table of the type seen in component specifications.

According to an additional aspect of the invention, the ECAD system includes a sub-system for gathering and condensing (preferably all) information relevant to a design. This information may include information about the origin of a design, the various objects which make up the design, the *revision history of the design*, and the authorship of the design. The information may also include information concerning the hardware specifications of the components of the design, for example, circuit loading, performance and timing information, and *drive strength*.

The various pieces of information available to the designer may typically already exist in diverse areas of an ECAD system--for example in, a *schematic editor*, a logic compiler, and a simulation program, as well as in various databases. These areas, or sources, are generally well known tools operating on an ECAD system. The present invention provides a user interface for managing retrieving information from such diverse sources.

According to an additional aspect of the invention, a user interface is provided with the capability of displaying abstract representations of a design in progress. These representations include all levels of design abstraction and include, but are not limited to, system level, architectural level, RT level, logic level, gate level and transistor level. The user interface also is also provided with the capability of linking like kinds of information over different representations, for example, a block diagram view and a gate level implementation of the block diagram. Common signal paths are shown in the block diagram and gate level views using a common symbology, such as color, shading, dashed lines, or the like. Alternatively, one view (e.g., block diagram) may be superimposed on another view (e.g., schematic)."

Col. 40, lines 18-56 disclose::

"FIG. 26 is a schematic illustrating the correlation of like kinds of information from a representation of a design at one level of design abstraction to the representation of the same design at a different level of design abstraction. The screen of the Graphical Display Device 1606 contains, by way of example, two windows 2805 and 2701. In the example of FIG. 26, window 2805 contains a Architectural Level view of a circuit. Window 2701 is a logic level view of the same circuit shown is

window 2805. The circuit in window 2805 contains registers 2703 and 2705. These correspond to a series of flip-flops, respectively. Datapath 2707 leading into register 2703 is the same datapath as datapath 2708. To enable the user to readily ascertain that these datapaths are the same, paths 2707 and 2708 are each displayed as a dashed line. Similarly, paths 2709 and 2710 refer to corresponding representation of another datapath displayed in windows 2805 and 2701, respectively. In order to visually indicate correspondence of the paths 2709 and 2710, they are each displayed with the same line style, different from the line style used for paths 2707 and 2708 (in this case, a dashed line adjacent to a solid line). In alternative embodiments, like signal paths can be displayed using color-based coding or by using other types of graphical coding to indicate commonality.

Additionally, selecting an object in one window will cause that object to be automatically highlighted in all other windows in which it appears. In FIG. 26, a cursor 2601 is shown pointing to flip-flop 2712. The user has caused the selection of flip-flop 2712, thereby causing it to be high-lighted. The selection of flip-flop 2712 in turn causes a section 2711 to be highlighted (shaded, in the figure) in block 2705 of window 2805. Because of the scale differential between the two representations, the highlighting in window 2805 may be displayed as the lighting or flashing of a very small dot inside of box 2705. Alternatively, the smallest object in the higher level representation, which contains the selected object in the lower level representation, is highlighted in its entirety.

Also shown in FIG. 26 is menu bar 2821, which is described in greater detail below with respect to FIG. 28.

FIG. 27 is a schematic utilizing overlays to illustrate the different levels of design abstraction on the same display. Window 2805' on Graphical Display Unit 1606 contains a circuit at the block diagram level of abstraction. It contains, among other elements, registers 2703' and 2705'. Window 2701' contains a portion of the same circuit at the gate level. It contains two series of flip-flops shown in the interior of boxes 2704 and 2706, corresponding to register 2703' and register 2705', respectively, and indicating direct correspondence between the register 2703' in window 2805' and the "boxed" flip-flops 2704 in window 2701', and between the "boxed" flip-flops 2706 in window 2805'. A common shading style is used to visually highlight the corresponding boxes (2704 and 2706) and registers (2703' and 2705', respectively)."

See, also: abstract; fig. 2-3, 6-10, 12-13, 17, 23-24, 26-27, 32-45.

Response to Arguments

13. Applicant's arguments filed 5/26/2000 have been fully considered but they are not persuasive.

14. Applicant's arguments filed 5/26/2000 have been fully considered but they are not persuasive. Representative has objected to the last Official Office Action as "*legally inadequate*" (*although no legal precedence has been cited*) and "*clearly erroneous factually*" (pp. 3-4, paper # 5). Examiner respectfully disagrees. The Examiner also objects to the tone expressed throughout paper # 5 including, for example, characterizing the prior art rejections as "*hopelessly inadequate*" (page # 5). In response, Examiner *respectfully* requests that Representative reconsider the claims and critically review the art as well as the associated class and subclasses. The Examiner is only interested in the *patentability* of the *claims* and will be persuaded by *factual* and *detailed arguments*.

15. In any case, the present Examiner *fully agrees* with the previous Examiner concerning the *prior teaching of Applicant's claims in the art* and the *appropriateness of the applied art rejections*.

16. As per remarks (page 4, paper # 4) concerning the Nishiyama rejection, Representative has provided no substantial evidence showing that Nishiyama does not teach user manipulation of design database elements other that the reference to fig. 1. The reasoning that Nishiyama discloses an *automatic* feature somehow teaches away is simply not convincing. In any case, Representative's various arguments concerning the art rejection are moot in view of the new art and other rejections. Examiner has also provided more detail concerning the original art rejection.

17. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dr. Hugh Jones whose telephone number is (703) 305-0023.

Dr. Hugh Jones

June 25, 2000

ERIC W. STAMBER
PRIMARY EXAMINER